REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
10-11-2010	Technical Paper	NOV 2010 - DEC 2010

4. TITLE AND SUBTITLE

Efficient Transmission of DoD PKI Certification in Tactical Networks

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sean R. O'Melia, Roger I. Khazan, and Dan Utin

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force
GG-12 CPSG/ZX
240 Hall Rd
San Antonio, TX 78243

10. SPONSOR/MONITOR'S ACRONYM(S)

GG-12 CPSG/ZX

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In tactical networks, transmission of DoD PKI digital certificates can create unnecessary burden on low-bandwidth links, increase response time for users, and drain radio power. In this paper we present a simple and practical approach to alleviating this problem. We develop a DoD PKI-specific compression dictionary that can be used to prime general purpose compression of certificates, resulting in a significant reduction of certificate sizes. We evaluate this approach using a sizable and diverse dataset of real DoD PKI certificates. Our evaluation suggests that the transmission and storage sizes of DoD PIG certificates can be reliably reduced by about 50%.

15. SUBJECT TERMS

Public key infrastructure, constrained networks, data compression

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
U					Zach Sweet
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	SAR	8	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			781-981-5997

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

MS-463794

Efficient Transmission of DoD PKI Certificates in Tactical Networks*

Sean R. O'Melia
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA, USA
sean.omelia@ll.mit.edu

Roger I. Khazan
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA, USA
rkh@ll.mit.edu

Dan Utin
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA, USA
danu@ll.mit.edu

ABSTRACT

In tactical networks, transmission of DoD PKI digital certificates can create unnecessary burden on low-bandwidth links, increase response time for users, and drain radio power. In this paper we present a simple and practical approach to alleviating this problem. We develop a DoD PKI-specific compression dictionary that can be used to prime general-purpose compression of certificates, resulting in a significant reduction of certificate sizes. We evaluate this approach using a sizable and diverse dataset of real DoD PKI certificates. Our evaluation suggests that the transmission and storage sizes of DoD PKI certificates can be reliably reduced by about 50%.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security, Performance

Keywords

Public key infrastructure, constrained networks, data compression

1. INTRODUCTION

Department of Defense (DoD) Public Key Infrastructure (PKI) underlies and enables much of the department's secure network operations. As part of this infrastructure, individuals and devices are issued long-term digital certificates. Existing and emerging DoD applications and protocols rely on such certificates to protect communication of individuals and devices, and to control their access to data and services. Consequently, applications and services frequently communicate DoD PKI certificates.

In tactical networks, transmitting digital certificates can create unnecessary burden on low-bandwidth links, increase response time for users, and drain radio power. In this paper we present a simple and practical approach to alleviating this problem. Our approach exploits the fact that DoD PKI certificates are minted according to a handful of common profiles, resulting in a lot of redundancy across different certificates.

Our approach uses general-purpose data compression, applied to certificates either at the application level or at a network proxy. However, simply compressing individual certificates, while providing some benefit, does not take advantage of the vast redundancies present across DoD PKI certificates. Our idea is to take advantage of these redundancies by priming general-purpose data compression with a special pre-placed dictionary that contains representative elements from DoD PKI certificates.

Our results demonstrate that such a dictionary, though constructed from a small sampling of DoD PKI certificates, can significantly improve compression of other DoD PKI certificates drawn from across the entire DoD. Specifically, the sizes of current DoD PKI certificates are reduced from about 1077 bytes to 500-700 bytes. Emerging NSA Suite B [17] certificates that are based on Elliptic Curve Cryptography (ECC) are reduced by 55% on average from 828 bytes to only 370 bytes.

Given that certificates are communicated frequently by applications and services, such reductions of certificate sizes can result in significant aggregate benefits for wireless links. In section 2 we present a number of motivating examples where such savings are important.

Our contributions can be summarized as follows:

1. We present results of a carefully designed experiment that studies the effects of data compression on DoD PKI certificates, using a sizable and diverse dataset of real DoD PKI certificates.

2. We establish the feasibility of developing and standardizing a relatively small DoD PKI-wide dictionary that, when used with general-purpose data compression, can significantly reduce the size of DoD PKI certificates, drawn from across the DoD.

3. We developed a generic software utility for constructing a compression dictionary from a sampling of digital certificates. The resulting dictionary contains common

*This research was sponsored by the United States Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are not necessarily endorsed by the US Government.

elements found in multiple certificates and is useful in priming compression of other, similar certificates.

Outside of demonstrating benefits of a standard DoD PKI-wide compression dictionary, our approach is ideally suited for focused tactical applications that use special-purpose PKI, such as future secure Small Unmanned Aerial Systems (SUAS).

Though this paper focuses on DoD PKI, note that the presented approach of using generic data compression and a domain-specific pre-placed dictionary applies to many other standard message types used in the DoD, such as Net-Centric Enterprise Services (NCES) [16], Cursor on Target (CoT) [2], and the emerging DoD's Key Management Infrastructure (KMI) [10]. Quantifying the benefits of our approach applied to these other domains is outside the scope of this paper and is subject to possible future work.

After presenting several motivating examples in the next section, the rest of the paper is organized as follows: In section 3 we provide relevant background about DoD PKI and data compression. In sections 4 and 5 we describe the experiment set-up and analysis results. In section 6 we also analyze compressibility of Certificate Revocation Lists (CRLs) and of the emerging NSA Suite B Elliptic Curve Cryptography certificates. In section 7 we discuss practical deployment of compression aided by pre-placed dictionaries. In section 8 we conclude.

2. MOTIVATING EXAMPLES

Below we draw examples from a wide range of tactical situations that involve frequent use and transmission of digital certificates: from controlling access to unmanned aerial vehicle (UAV) video feeds, to secure session establishment in tactical networks, to access verification for web services, to secure email messaging, to bulk certificate transfer to forward-deployed units.

2.1 Secure UAV Operation

Consider a scenario where warfighters' remote video terminals (RVTs) need to authenticate to UAVs[1] in order to be granted access to the UAVs' video feeds. In the near future such authentication will likely be accomplished with certificates. Since the uplink from an RVT to a UAV is low bandwidth, and both devices are battery-powered, reducing the certificate size with compression prior to its transmission would result in faster response time for the warfighter and better utilization of the batteries.

Another emerging concept is to use small UAVs (SUAVs) as routers to enable communication among warfighters on the ground, in the air, and at sea. For example, warfighters may utilize SUAVs for sending text chat messages to each other or jointly annotating and interacting over a shared map. Similarly to the above example, authentication and access control will be accomplished with digital certificates, which will have to be transmitted by warfighters' terminals up to SUAVs via low-bandwidth and power-constrained links.

2.2 Tactical Net-Centric Operations

More generally, outside of the UAV domain, the vision of network-centric warfare (NCW) calls for collaboration and information sharing among warfighters.

For warfighters at the tactical edge, a part of this collaboration and information sharing is accomplished by accessing DoD websites and web services via standard applications, interfaces, and protocols over constrained radio links. All of such accesses involve transmissions of DoD PKI certificates during session establishment. For example, every secure email message carries the sender's PKI certificate, or every time a warfighter accesses a DoD website, there is an exchange of certificates between the warfighter's web-browser and the DoD's web-server.[2]

As another example, mobile ad-hoc networks (MANETs) play a considerable role in the envisioned NCW operations. Security of MANETs naturally relies on frequent transmission of certificates for verifying identities of participating MANET nodes.

2.3 Bulk-transfers of certificates in-theater

Forward-deployed, tactical units that execute missions without real-time connectivity to the global DoD network typically rely on daily transfers of information via a satellite link to their network-connected system(s). As part of such transfers, certificates and CRLs that are relevant to upcoming missions are likely to be transferred in bulk.

In summary, there are many compelling situations that highlight the importance of reducing overhead of certificate transmissions in tactical networks.

3. BACKGROUND

In this section we summarize relevant information about DoD PKI and data compression.

3.1 DoD PKI

Confidentiality, integrity, and authenticity of data are crucial to the security of DoD information operations. Public key infrastructure has been established to facilitate secure electronic communications between entities throughout the DoD [8]. In general, PKI applies asymmetric cryptography to protect information disseminated among members of an organization. It also provides facilities to obtain and verify users' information in a directory. The most important components of PKI from a user perspective are digital certificates. A digital certificate is a data structure that binds a public cryptographic key to an identity. The public key is typically based on a well-known asymmetric cryptographic algorithm such as the Rivest-Shamir-Adleman (RSA) algorithm [21]. Identity information consists of a number of *metadata* fields describing the subject entity. The public key and identity metadata are bound together through the use of a digital signature [15], applied by the Certificate Authority (CA) that issued the certificate. CAs themselves possess their own certificates for verifying their identities that are in turn signed by higher-level CAs. This so-called "chain of trust" extends up the PKI hierarchy to a Root CA at the topmost level and is intended to provide assurance of the authenticity of a digital identity.

Digital certificates in PKI are structured in accordance with the X.509 standard [13] created by the International Telecommunication Union's Standardization Sector (ITU-T). A general profile for X.509 certificates is specified in

[1] or to their controlling stations via the UAVs' comm channels

[2] A study performed by Lincoln Laboratory staff (to be published) shows that certificates comprise the majority of network traffic transmitted during a secure web session.

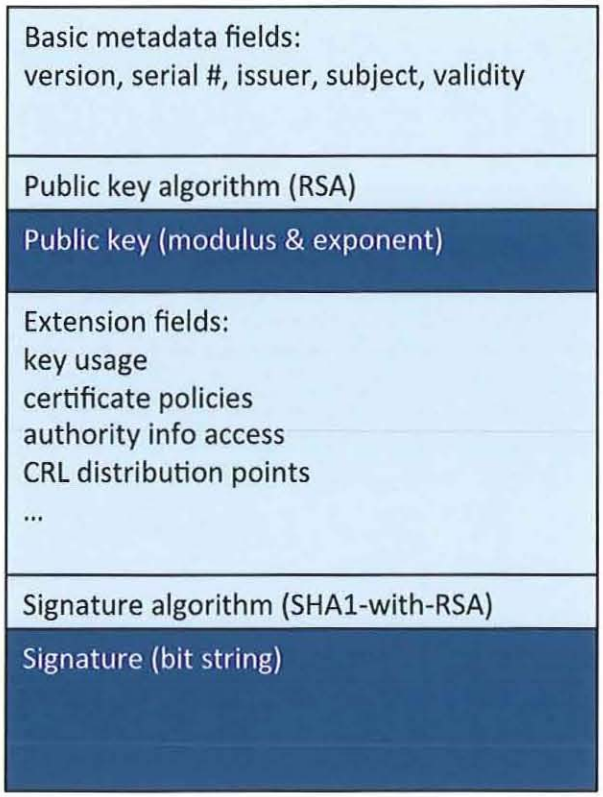

Figure 1: Sample profile for RSA-based certificates within DoD PKI. Lightly shaded areas indicate certificate metadata that is mostly human-readable.

RFC 5280 [3], published by the Internet Engineering Task Force (IETF). Certificates in the X.509 format are required to contain a number of *basic fields*, including version, serial number, issuer, subject, validity, public key, and the algorithm associated with the public key. Starting with version 3, X.509 certificates may have zero or more *extension fields*, specifying for example key usage, certificate policies, authority information access, and subject and authority key identifiers.

In this paper, we use the term *profile* to refer to the format structure of a given certificate, specifying the types of the metadata fields and the cryptographic algorithms. Several extension fields, including Authority Key Identifier, Certificate Policies, and Authority Information Access, contain values common among all certificates originating from the same CA.

Fig. 1 illustrates a profile for RSA-based digital certificates issued by DoD CAs [23]. Whereas the individual fields are not detailed in the figure, we emphasize the distinction between human-readable text in the metadata fields and the random-appearing bytes in the public-key and digital-signature fields. The sizes of labeled sections in the figure are approximately proportional to their sizes in an actual certificate. Certificates constructed using this profile would contain about 63 percent human-readable metadata and 37 percent random-appearing data in the public-key and signature fields.

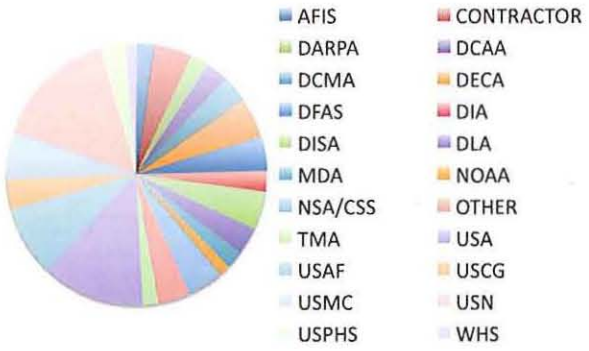

Figure 2: Distribution of DoD organizations in our dataset

3.2 Data Compression

The purpose of data compression is to decrease the size of given data by removing redundancies within it. Data compression may be either *lossless*, where data must be decompressible into its original form, or *lossy*, in which case some fidelity with respect to the original data may be lost. When compressing digital certificates for storage and transmission in any environment, lossless compression is essential in order to preserve the integrity of the certificate data. Many lossless compression libraries exist and most could have been used in this project.

We chose the zlib compression library [25], which is free from patents and implements lossless compression [5, 4] that is independent of CPU architecture, operating system, and file system. The zlib library combines the Lempel-Ziv compression algorithm [24] and Huffman coding [12], resulting in compression performance that is on par with the best compression methods. In the worst case where the input data cannot be effectively compressed, it is expanded at a ratio of five bytes per 16-kilobyte block, or 0.03 percent.

The zlib library includes a feature for initializing the (de)-compression processes with a *pre-placed dictionary*; compression performance improves when the dictionary contains byte strings that are likely to appear in the input data to be compressed/decompressed. The compressor and decompressor must be initialized with the exact same dictionary; otherwise, the decompressor will not be able to properly recover the data.

4. EXPERIMENTAL STUDY

In this section, we overview first our DoD PKI certificate dataset and then our approach for generating a pre-placed dictionary to aid DoD PKI certificate compression.

4.1 Certificate dataset

All certificates in our dataset were retrieved through the DoD Global Directory Service [7]. By design, the dataset contains a representative cross-section of the DoD PKI certificate space, ranging across different CAs and organizations. Specifically, as shown in Fig. 2, our dataset represents 22 different Organizational Units as specified in each certificate's `Subject` field. The armed forces, including Army,

Air Force, Coast Guard, Marine Corps, and Navy represent about half of the entire certificate dataset.

Also by design, the collection of certificates to be used for pre-placed dictionary generation is separated in the dataset from that to be used for compression testing. The former one consists of 192 certificates with the last name **Smith**; the testing collection consists of 170 certificates with the last name **Johnson**. Both collections have been assembled in a way to ensure that our experimental results would translate to similar results if our approach were to be deployed in real tactical environments.

Our investigation of the collected certificates revealed that all certificates specify Key Encipherment in the `KeyUsage` extension field and are signed by a DoD E-mail CA, with the `SignatureAlgorithm` being RSA-SHA-1 [23]. These fields indicate that the certificates are intended for use with secure e-mail.

Furthermore, we found that many certificates have very common structure across different organizations, and across many CAs. We were able to group all of the certificates into four profiles: A, B, C, and D.

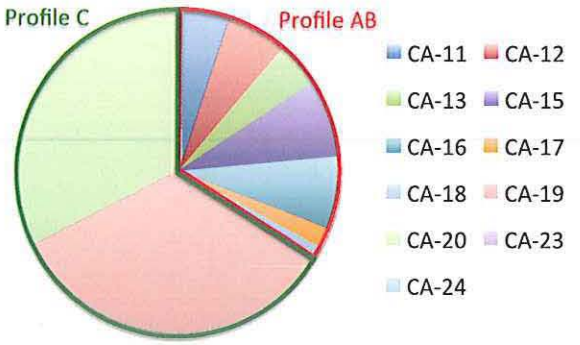

Figure 3: Distribution of Certificate Authorities in our dataset

Certificates is Profile A and Profile B include a 1024-bit RSA public key and 1024-bit signature [23], [9]; their metadata fields comprise about 75 percent of the total certificate size. The sole difference between these two profiles is the presence of an additional 20-byte-long metadata field in Profile B, called `SubjectDirectoryAttributes`. Because of their similarity, we combined the Profile A and Profile B certificates into one category: Profile AB. Profile AB has 121 certificates, averaging 1077 bytes in size.

Certificates in Profile C are structured somewhat differently: each certificate contained a 2048-bit signature [23], [9] and its metadata fields comprise about 63 percent of the total certificate size. The smaller amount of metadata in this profile offsets the larger digital signature, leading to an average certificate size of 1077 bytes, same as in Profile AB. Profile C is the dominant category in our dataset (237 certificates).

Profile D certificates have structure very similar to Profile A except for public key and signature sizes of 2048 bits each [23], [9]. Such certificates are only starting to be used in the DoD PKI [23]. We were unable to locate a sufficient number of certificates in this profile (only 4 out of 362 collected), and as the result excluded Profile D certificates from the study.

Fig. 3 illustrates the distribution of CAs throughout our dataset and highlights which CAs use Profile AB and which use Profile C. The ratio of Profile AB certificates to Profile C certificates throughout the entire dataset was 33.8 percent to 66.2 percent.

4.2 Dictionary creation

In order to evaluate how much savings in storage and transmission can be achieved by applying dictionary-aided compression to digital certificates, we designed and conducted several compression experiments with differently constructed dictionaries. The resulting sizes of these compressed certificates were compared to their uncompressed sizes and to their compressed sizes under the case where no pre-placed dictionary was in use.

A naive approach: The most basic approach for creating the dictionary is to place complete certificates in the dictionary with the idea that their substrings are likely to appear in other certificates to be compressed. This approach has two major flaws. First, since the compressor's memory buffer is limited to 32 kilobytes of data, such a dictionary would contain only about 30 certificates before the buffer is completely filled, thus making it impossible to have broad representation of entities across the DoD. Second, a significant amount, though not a majority, of a certificate's data is comprised of the public key and digital signature, which have no similarity with other certificates and hence would not help compression.

A better approach: Better use can be made of the available memory buffer space by simply extracting metadata from several certificates and constructing a dictionary by concatenating these substrings. Whereas this approach admits more useful data into the initialized memory buffer, there are still some metadata fields whose values are unique to an individual certificate or a very small number of certificates. These include, but are not limited to, the `CommonName` portion of the `Subject` field, `SubjectKeyIdentifier`, `SubjectAlternativeName`, and even the validity fields (`NotBefore` and `NotAfter` time-stamps). Since these particular substrings will have low chance of occurrence in other certificates, they can be excluded from the dictionary. Furthermore, some certificates may share a relatively large amount of common data, usually related to a common CA, such as `Issuer` (including `CommonName`), `AuthorityKeyIdentifier`, `AuthorityInfoAccess`, and `CRLDistribution- Points`. These duplicate substrings need only be represented once in the dictionary. These fields are expected to aid compression significantly since they reflect redundancies among groups of certificates from common origins.

In this paper we report the results of using a pre-placed compression dictionary that contains unique values of frequently occurring certificate metadata. Since it is expected that random-looking data will not compress well, we also investigated the effect of dictionary-aided compression specifically on the portions of certificates that would actually be compressible; that is, the various metadata fields contained in the certificates.

4.3 Software Tools

We used zlib (v1.2.3) for all data compression operations. A simple wrapper program was written in C to specify target files to be compressed as well as dictionaries on the com-

mand line. To extract individual fields from certificates, we used the X.509 subset of the C API included with OpenSSL (v1.0.0-beta2) [18]. Dictionary generation and bulk certificate compression tests were handled through scripting in bash and Perl. All tests were performed on a MacBook Pro running Mac OS X (v10.5.8).

5. RESULTS AND ANALYSIS

Results of the compression tests, graphed in Figs. 4 and 5, show the average sizes for the indicated profiles in uncompressed form, compressed form without a pre-placed dictionary, and compressed form with the aid of a pre-placed dictionary, as described in section 4.2. In each graph, the left column represents the 51 certificate test samples in Profile AB, and the right column represents the 118 samples in Profile C. The lower portions of each column show the mean results of compressing **Johnson** certificates in Profile AB (C) using dictionaries created from **Smith** certificates in Profile AB (C). Error bars for each data series indicate one standard deviation above and below the mean. The small sizes of the error bars indicate very little variation in certificate sizes and compression results across the dataset.

Fig. 4 shows results for full certificates. Profile AB certificates were on average compressed from 1077 bytes to 829 bytes without the dictionary, and to 532 bytes with the dictionary. As for Profile C certificates, compression without the dictionary led to a size reduction from 1077 bytes to 887 bytes, and dictionary-aided compression reduced the average certificate size to 681 bytes. It can be seen that certificates in Profile AB have better compression results than those in Profile C, both with and without the dictionary. This is due to the fact that Profile AB certificates have a greater proportion of metadata than Profile C certificates. The graph also shows that compression with the aid of a dictionary achieves approximately twice the space savings achieved when compressing certificates without a pre-placed dictionary.

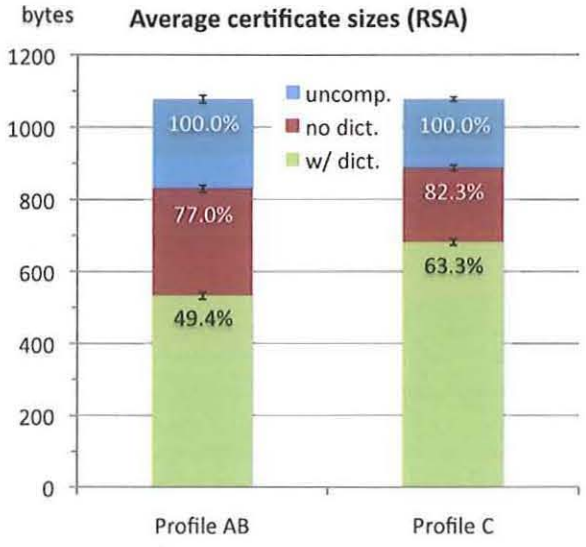

Figure 4: Results of compression for DoD PKI dataset, full certificates

Fig. 5 shows the mean results for compressing only the metadata extracted from certificates. For the Profile AB certificates, metadata was on average compressed from 809 bytes to 546 bytes without the dictionary, and to 250 bytes with the dictionary. In Profile C, compression of metadata without the dictionary led to a size reduction from 681 bytes to 469 bytes, and dictionary-aided compression reduced the average certificate size to 264 bytes. Uncompressed metadata as well as metadata compressed without the dictionary are larger for Profile AB than Profile C due to the higher amount of metadata in the Profile AB category. In relative terms, the percentages resulting from compression without the dictionary are about equal—67.5 percent for Profile AB compared to 68.9 percent for Profile C. However, when compressing with the aid of the dictionary, the absolute sizes in bytes of compressed metadata for each category are approximately equal. Thanks to the pre-placed dictionary, most of the certificate metadata has been compressed out and converted to their encoded representations.

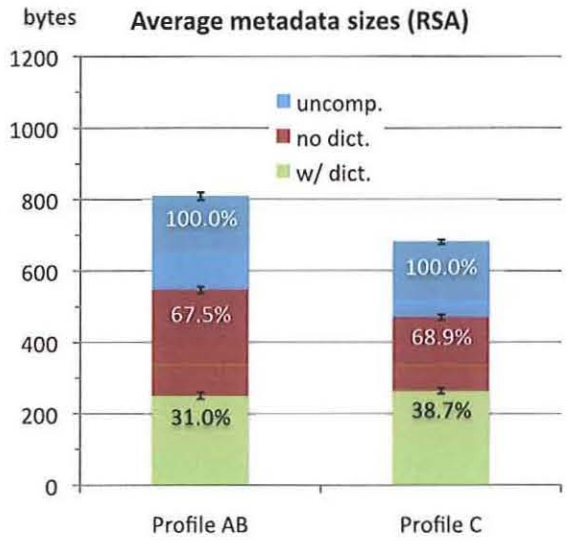

Figure 5: Results of compression for DoD PKI dataset, certificate metadata only

The above experiments establish the compression results that are likely to be observed in the real world when the profiles of the certificates being compressed are represented in the pre-placed dictionary; recall that we observed that current DoD PKI certificates seem to all fit into a handful (~ 4) of profiles.

For completeness, we also conducted a smaller-scale experiment to mimic a situation in which the profiles of the certificates being compressed are not represented in the pre-placed dictionary: We compressed certificates from Profile C using the pre-placed dictionary from Profile AB, and vice versa. The results are presented in Table 1, along with the results for using no pre-placed dictionary and using the same-profile dictionaries. As expected, compression of the certificates whose profiles are not represented in the pre-placed dictionary leads to poorer results compared to using the dictionary that includes the certificates' profile. However, these results also show that using a wrong pre-placed

Table 1: Effects of cross-profile dictionary compression on certificate size (all figures are in bytes)

		Dictionary		
Dataset	Uncomp.	none	AB	C
AB, full cert.	1077	829	532	665
AB, metadata	809	546	250	382
C, full cert.	1077	886	717	681
C, metadata	681	469	299	264

dictionary can still be better than not using one at all.

6. OTHER PKI STRUCTURES

6.1 Certificate Revocation Lists

We briefly explored other types of X.509 structures expected to be disseminated in DoD PKI communications. One type is the certificate revocation list (CRL), containing serial numbers and dates of invalidation for certificates revoked by a particular CA. We tested the effect of zlib compression on a set of CRLs from [7] and found that the compressed aggregate size was only about 32 percent of the uncompressed aggregate size. Since the individual CRL records contain identical ASN.1-specific encodings [14], the compression algorithm is able to take advantage of a large amount of redundancy within a single CRL, leading to good compression performance, even without a pre-placed dictionary.

6.2 ECC certificates

Certificates currently issued by DoD rely on the RSA algorithm. Going forward, DoD has a strong interest in transitioning PKI to elliptic curve cryptography (ECC) [17], [10], [23]. ECC requires smaller keys than RSA for the same level of security because the types of arithmetic shortcuts that can be applied to attacking RSA are not available in ECC: 160-bit ECC keys give approximately the same level of security as 1024-bit RSA keys. The reduced certificate sizes that follow from the use of ECC can alleviate demands placed on constrained networks, such as those where SUAS are deployed.

Since real ECC certificates are not yet available from DoD PKI, we generated a set of ECC device certificates according to the standard NSA Suite B device certificate profile [23], [22], illustrated in Fig. 6. Each certificate was 828 bytes in size on average and consisted of about 692 bytes (84 percent) metadata with the rest accounted for in the public key and signature. Even at these uncompressed sizes, the ECC certificates consume as much space as compressed RSA-based certificates with 2048-bit keys, and have greater strength in terms of "bits of security" [1].

We used our collection of ECC certificates to generate a pre-placed dictionary and tested compression of the certificates using this dictionary, similarly to our main study. The results are shown in Fig. 7. The certificates compressed to about 649 bytes without a pre-placed dictionary, and to about 370 bytes with the dictionary. The metadata in our ECC device certificates compressed to 506 bytes without the pre-placed dictionary, or to 227 bytes with the dictionary. Compression rates in terms of percentage reduction are similar to those achieved for the Profile AB RSA certificates.

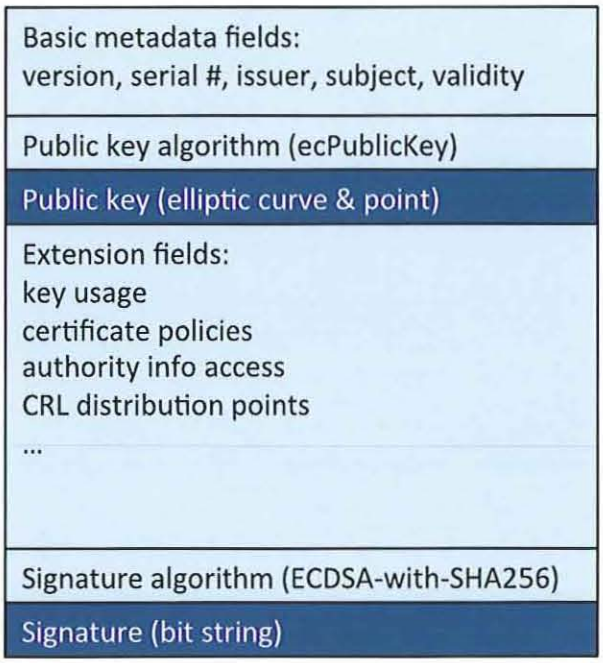

Figure 6: Sample profile for NSA Suite B ECC-based device certificates. Lightly shaded areas indicate certificate metadata that is mostly human-readable.

7. PRACTICAL DEPLOYMENT OPTIONS

In this section we discuss our considerations for practical deployment of pre-placed dictionaries for general data compression in various operating scenarios. These deployment options are drawn across four dimensions. First, the contents of the pre-placed dictionary may be targeted to a specific domain or application, e.g. SUAS communications using NSA Suite B device certificates. Alternatively, it could be a standard dictionary for the entire DoD PKI.

Second, the point where (de)compression is applied in the system depends on the structure of the network and capabilities of participating nodes. For example, the dictionary could be deployed at the application level on the host system. We are concerned primarily with disadvantaged networks composed mainly of power- and bandwidth-constrained systems. Under these circumstances, a better option may be to deploy the dictionary at a gateway node; an example is an *intercepting proxy* surrounding a disadvantaged communication link.

Third, any pre-placed dictionary to be used for data compression should be known to all participating elements in the network. In domain- or application-specific operating environments, a static dictionary could be pre-installed onto systems. In more dynamic environments, dictionaries may need to be transferred prior to a communications session through either a "push" operation by the session initiator or a "pull" operation by the participating nodes using a uniform resource indicator (URI) to obtain the dictionary. Another possibility is the negotiation of which pre-placed dictionary to use along with any associated parameters. In this scenario, multiple pre-placed dictionaries are present on the

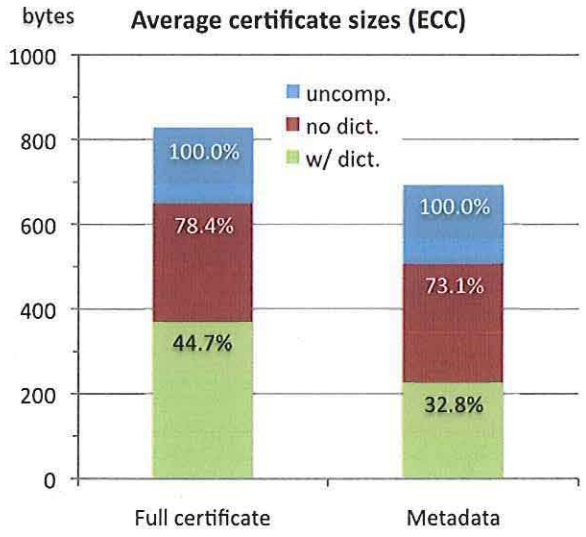

Figure 7: Results of compression for ECC device certificates

participating nodes, and the dictionary and version negotiation could be integrated with already existing negotiations for the application/protocol in use. For example, Transport Layer Security (TLS) [6] and Extensible Messaging and Presence Protocol (XMPP) [20] already include negotiation of compression capabilities as part of their session establishment. These can be extended to include pre-placed dictionary negotiation.

Fourth, we consider how dictionaries and the messages (de)compressed using these dictionaries may be formatted. Several messaging protocols utilize Cryptographic Message Syntax (CMS) [11] as the standard message format. CMS is highly extensible and specifies a `CompressedData` message structure [19]. This structure could be extended with a parameter such as a URI specifying a particular pre-placed dictionary. We will consider for future work the specification of pre-placed dictionary based compression for this and other standard formats and submission of the developed proposals as standards to the IETF.

8. CONCLUSIONS

As public key infrastructure becomes an increasingly important part of DoD information security, the problem of storage and transmission overhead, especially in constrained network environments, must be addressed. We have shown general data compression to be an effective tool for reducing size overhead associated with DoD PKI certificates. Compression aided by a pre-placed dictionary provides further improvement to compression rates and thus can be especially useful for constrained communication environments.

To summarize our results, we present in Table 2 average uncompressed and compressed (with dictionary) data sizes for certificates using RSA and ECC cryptography. All figures are in bytes. The top two rows for the RSA algorithm reflect our main experiments described in sections 4 and 5, and the third row shows figures based on our analysis of NSA Suite B ECC device certificates. The results demonstrate that using a pre-placed compression dictionary leads to significant size reductions for all certificate types explored. Depending on the type of certificate, our methods can achieve compression rates of one-third to one-half space savings.

Table 2: Results of dictionary-aided compression on certificates using common choices for public-key algorithms and sizes

	Full certificate		Metadata	
Profile	uncomp.	comp.	uncomp.	comp.
RSA Profile AB	1077	532	809	250
RSA Profile C	1077	681	681	263
ECC device	828	370	692	227

The intended future transition of DoD PKI from RSA to ECC is motivated in part by shorter keys and certificates. The compression-based approach presented here is complementary to this transition: ECC reduces the sizes of public keys and signatures in certificates, while compression with a pre-placed dictionary reduces the sizes of the metadata fields in the certificates. The additional byte reduction thanks to our approach is on par with the byte reduction obtained by moving from RSA to ECC.

Acknowledgment

The authors would like to thank Adam Petcher, Jon Herzog, Joe Cooley, Ben Fuller, and Richard Lippmann for their helpful contributions and comments towards this work.

9. REFERENCES

[1] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. NIST Special Publication 800-57: Recommendation for Key Management — Part 1: General. `http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf`, Mar. 2007.

[2] Creating Standards for Multiway Data Sharing. `http://www.mitre.org/news/the_edge/summer_04/harding.html`.

[3] D. Cooper et al. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. `http://tools.ietf.org/html/rfc5280`, May 2008.

[4] P. Deutsch and J.-L. Gailly. DEFLATE Compressed Data Format Specification version 1.3. `http://tools.ietf.org/html/rfc1951`, May 1996.

[5] P. Deutsch and J.-L. Gailly. ZLIB Compressed Data Format Specification version 3.3. `http://tools.ietf.org/html/rfc1950`, May 1996.

[6] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. `http://tools.ietf.org/html/rfc5246`, Aug. 2008.

[7] DISA Enterprise Directory Service – GDS & JEDS. `http://www.disa.mil/services/gds.html`.

[8] DoD Instruction 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling. `http://www.`

dtic.mil/whs/directives/corres/pdf/852002p.pdf, Apr. 2004.

[9] DoD PKE InstallRoot utility. https://www.dodpke.com/InstallRoot.

[10] DoD Public Key Infrastructure Program Management Office. Public Key Infrastructure Roadmap for the Department of Defense. http://iase.disa.mil/pki/dodpki-roadmap.doc, December 2000.

[11] R. Housley. Cryptographic Message Syntax (CMS). http://tools.ietf.org/html/rfc5652, Sept. 2009.

[12] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952.

[13] International Telecommunication Union. Recommendation X.509: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. http://www.itu.int/rec/T-REC-X.509-200508-I, Aug. 2005.

[14] International Telecommunication Union. Recommendations X.680–X.695: Information technology – Abstract Syntax Notation One (ASN.1). http://www.itu.int/rec/T-REC-X/e, Nov. 2008.

[15] National Institute of Standards and Technology. Federal Information Processing Standards Publication 186-3 — Digital Signature Standard (DSS). http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf, June 2009.

[16] Net-Centric Enterprise Services. http://www.disa.mil/nces.

[17] NSA Suite B Cryptography. http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml.

[18] OpenSSL: The Open Source toolkit for SSL/TLS. http://www.openssl.org.

[19] P. Gutmann. Compressed Data Content Type for Cryptographic Message Syntax (CMS). http://tools.ietf.org/html/rfc3274, June 2002.

[20] P. Saint-Andre (ed.). Extensible Messaging and Presence Protocol (XMPP): Core. http://tools.ietf.org/html/rfc3920, Oct. 2004.

[21] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[22] J. Solinas and L. Zieglar. Suite B Certificate and Certificate Revocation List (CRL) Profile. http://tools.ietf.org/html/rfc5759, Jan. 2010.

[23] United States Department of Defense X.509 Certificate Policy Version 10.0. http://iase.disa.mil/pki/dod_cp_v10_final_2_mar_09_signed.pdf, Mar. 2009.

[24] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.

[25] zlib Home Site. http://www.zlib.net.